Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №2**

**«Изучение одномерных массивов и строк»**

**ПО ДИСЦИПЛИНЕ «Основы Алгоритмизации и Программирования»**

Выполнил: студент учебной группы

ИСПк-204-52-00

Конышев Иван Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

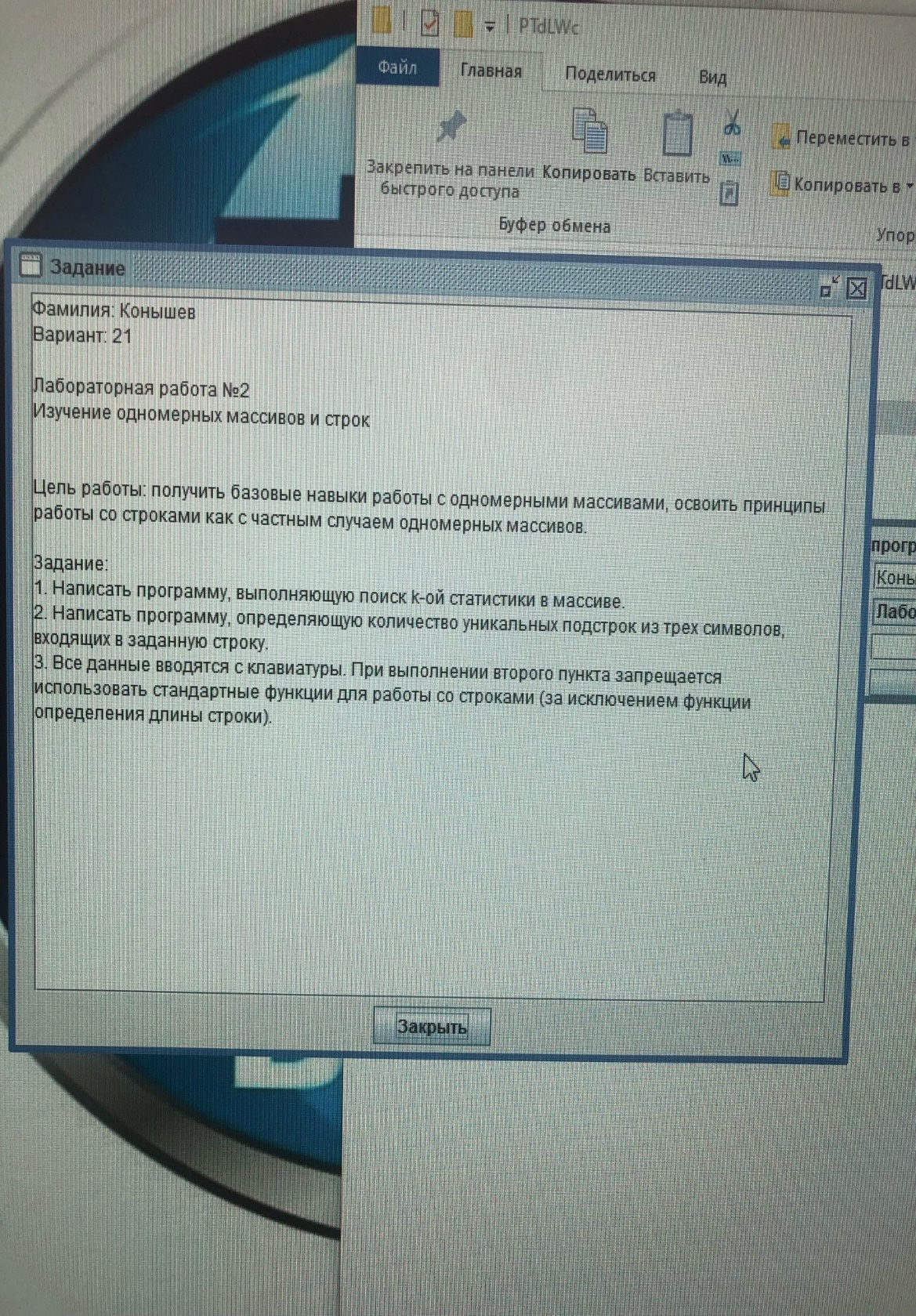
Киров

2023

**Цель работы**

Цель работы: получить базовые навыки работы с одномерными массивами, освоить принципы работы со строками как с частным случаем одномерных массивов.

**Формулировка задания**

****

**Описание алгоритма**

Данный алгоритм предназначен для поиска элементов в массиве, удовлетворяющих условию: произведение двух элементов не должно превышать их сумму.

1. Объявляется константа MAX\_SIZE, определяющая максимальный размер массива, и тип TArray, представляющий массив целых чисел.
2. Объявляется процедура QuickSort, которая сортирует массив arr методом быстрой сортировки. Она принимает параметры left и right, определяющие границы сортируемой части массива.
3. В процедуре QuickSort происходит разделение массива на две части: элементы, меньшие опорного элемента, и элементы, большие опорного элемента. Затем рекурсивно вызывается QuickSort для обеих частей массива.
4. Функция FindKthStatistic принимает массив arr, размер n и значение k. Она вызывает процедуру QuickSort для сортировки массива и возвращает k-ый элемент отсортированного массива.
5. В основной части программы:
   * Объявляются переменные arr, n, k и i.
   * Пользователю предлагается ввести размер массива n.
   * Пользователю предлагается ввести элементы массива.
   * Пользователю предлагается ввести значение k.
   * Если k находится в допустимом диапазоне от 1 до n, вызывается функция FindKthStatistic и выводится результат.
   * В противном случае выводится сообщение о некорректном значении k.

**Схема алгоритма**

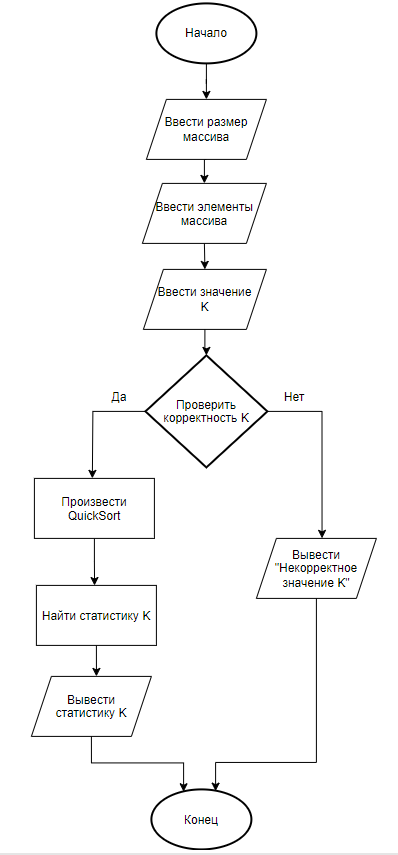
****

Рисунок 1 – Схема алгоритм задачи

**Код программы**

program zad1;

const

MAX\_SIZE = 100;

type

TArray = array[1..MAX\_SIZE] of Integer;

procedure QuickSort(var arr: TArray; left, right: Integer);

var

i, j, a, temp: Integer;

begin

if left < right then

begin

pivot := arr[left];

i := left;

j := right;

repeat

while (arr[i] < a) do

Inc(i);

while (arr[j] > a) do

Dec(j);

if i <= j then

begin

temp := arr[i];

arr[i] := arr[j];

arr[j] := temp;

Inc(i);

Dec(j);

end;

until i > j;

QuickSort(arr, left, j);

QuickSort(arr, i, right);

end;

end;

function FindKthStatistic(arr: TArray; n, k: Integer): Integer;

begin

QuickSort(arr, 1, n);

FindKthStatistic := arr[k];

end;

var

arr: TArray;

n, k, i: Integer;

begin

Write('Введите размер массива: ');

Readln(n);

WriteLn('Введите элементы массива:');

for i := 1 to n do

Readln(arr[i]);

Write('Введите значение K: ');

Readln(k);

if (k >= 1) and (k <= n) then

WriteLn('K-ая статистика в массиве: ', FindKthStatistic(arr, n, k))

else

WriteLn('Некорректное значение K.');

end.

**Результат выполнения программы**

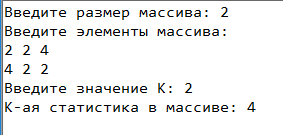


Рисунок 2 – Результат выполнения программы 1

**Описание алгоритма второго кода**

1. Объявление переменных

2. Инициализация множества uniqueSubstrings пустым множеством.

3. Вывод сообщения "Введите строку: " для запроса ввода от пользователя.

4. Считывание строки с клавиатуры и сохранение в переменную inputString.

5. Цикл for с переменной-счетчиком i от 1 до length(inputString) - 2. Цикл выполняется для каждой позиции в строке, начиная с первого символа и до третьего символа перед концом строки.

6. Внутри цикла:

* + Формирование трехсимвольной подстроки substring путем конкатенации символов inputString[i], inputString[i+1] и inputString[i+2].
  + Добавление подстроки substring в множество uniqueSubstrings с помощью операции объединения множеств [substring].

7. Вывод сообщения "Количество уникальных подстрок из трех символов: " с последующим выводом содержимого множества uniqueSubstrings.

**Схема алгоритма второго кода**

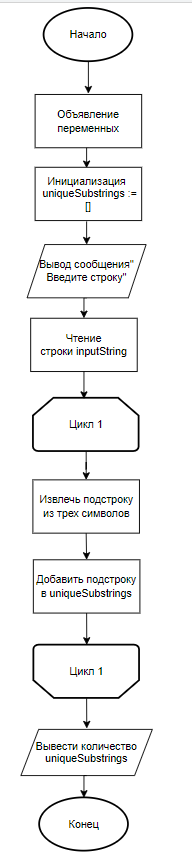


Рисунок 3 - Схема алгоритма задачи 2

**Код программы**

program zad2;

var

inputString: string;

uniqueSubstrings: set of string;

i: integer;

substring: string;

begin

uniqueSubstrings := [];

writeln('Введите строку: ');

readln(inputString);

for i := 1 to length(inputString) - 2 do

begin

substring := inputString[i] + inputString[i+1] + inputString[i+2];

uniqueSubstrings := uniqueSubstrings + [substring];

end;

writeln('Количество уникальных подстрок из трех символов: ', uniqueSubstrings);

end.

**Результат выполнения программы**

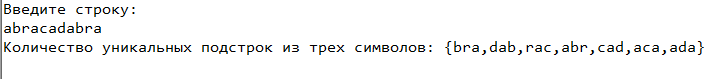


Рисунок 4 – Результат выполнения программы 1

**Вывод**

В ходе выполнения домашней лабораторной работы мы узнали, как реализовать программу на языке Pascal, которая сортирует массив целых чисел и находит k-ый элемент отсортированного массива. Таким образом, по рассмотренным задачам мы получили навыки работы с массивами, сортировкой, рекурсией, вводом-выводом данных и множествами в языке Pascal. Так же были использованы знания, полученные на лекциях и при выполнении лабораторных работ №5-6 и №7-8. Никаких сложностей при выполнения домашней контрольной работы не встретилось.

Так же для написания схемы алгоритма был использован интернет -ресурс «draw.io». Его использование упрощает написание схем алгоритмов, уменьшает вероятность ошибок, а также в случае пропуска какого-либо цикла его можно без каких-либо проблем вставить в любое место схемы.